

Целью работы является проведение исследования процесса построения экспертных систем в заданной предметной области с помощью пакета прикладных программ фирмы EXSYS Inc.

2. Теоретическая часть

Основой экспертной системы является база знаний. RuleBook - это средство разработки баз знаний и систем для их обработки, называемых здесь системами RuleBook .

2.1. Описание системы RuleBook

При старте системы выполняются следующие действия:

- а) сначала вводятся _____ -т.е. цели, по выбору которых система будет принимать решение;
- б) затем вводятся _____, которые будут использоваться, чтобы сделать выбор, они могут быть числовые или текстовые;
- в) потом вводится диаграмма в виде дерева, которая описывает, как принять решение по выбору альтернативы, основанное на различных комбинациях факторов;
- г) система проверяется с помощью Эксперта RuleBook. RuleBook будет автоматически проверять систему на наличие общих _____ и поможет вам устранить их.

Альтернативы

Альтернативы - цели, которые система будет выбирать в каждой развилке дерева решений. Количество альтернатив ограничено лишь параметрами ЭВМ, на которой создается эта экспертная система.

Факторы

Факторы - блоки информации, которые система будет использовать, чтобы выбрать цель. Факторы могут быть числовые или текстовые. Числовые величины фактора могут быть проверены алгебраически. Текстовые факторы (качества) представлены в системе в виде списков вариантов.

В RuleBook логика принятия решений представлена в виде диаграммы дерева решений. Каждый блок дерева является одним из факторов.

Факторы могут быть числовые в определенном диапазоне, выражения или качества. Каждая из возможных величин фактора рассматривается в блоке.

Например, левый блок (Фактор 1) имеет 3 возможных величины. Когда узел добавлен к дереву, может быть определена группировка величин в ветви.

RuleBook будет генерировать ветвь для каждой возможной величины (или группы величин). В этом случае каждая - отдельная ветвь. Фактор 2 и Фактор 3 блока каждый имеют две величины. Иметь каждый Фактор на каждой отдельной ветви необязательно.

Каждая ветвь дерева имеет две части - часть "ЕСЛИ" и часть "ТО". Узлы "ЕСЛИ" - условные выражения. Узлы "ТО" - присваивание величины фактору или выбору. В примере боксы альтернатив - части "ТО". Все "ЕСЛИ" узлы будут иметь, по крайней мере, 2 возможных величины. Все "ТО" узлы будут иметь только единственную величину. экспертных систем в заданной предметной области с помощью пакета прикладных программ фирмы EXSYS Inc. Узлы "ТО" могут также присвоить значение переменной или качеству. Ветви, которые устанавливают величины для факторов, используются в системах с обратным формированием цепочки ввода, чтобы получить данные от других информационных систем.

Каждая ветвь в дереве преобразуется в правило (rule). В RuleBook используются правила продукций вида "ЕСЛИ"... "ТО".

Пример правил в RuleBook :

Rule 1:

IF

Factor 1: Value 1

and Factor 2: Value 1

THEN

Choice 1

Rule 2:

IF

Factor 1: Value 1

and Factor 2: Value 2

THEN

Choice 2

Rule 3:

IF

Factor 1: Value 2

and Factor 3: Value 1

THEN

Choice 3

Rule 4:

IF

Factor 1: Value 2

and Factor 3: Value 2

THEN

Choice 4

Rule 5:

IF

Factor 1: Value 3

THEN

Choice 5

2.2. Эксперт

Эксперт RuleBook проверяет систему на наличие логических ошибок. Чтобы проверить вашу систему, щелкните кнопкой ПРОВЕРКА СИСТЕМЫ. Если эта кнопка имеет красную границу, правила не редактировались с момента последней проверки.

RuleBook сначала делает быстрый контроль системы. Если ошибки найдены при быстром контроле, RuleBook спрашивает, продолжать ли полную проверку. Полный контроль большой системы может занимать много времени, поэтому обычно если ошибки найдены при быстром контроле, лучше исправить их перед выполнением полного контроля. "Полное Испытание Системы" проверит все возможные комбинации входов.

RuleBook вычисляет максимальное количество запусков, которое потребуется. Поскольку некоторые качества и переменные зависят от других, фактическое количество проверок часто меньше максимального. Если число проверок для полного систематического контроля очень велико, RuleBook предложит вам вариант случайных испытаний. Случайное испытание не такое полное, как систематическое, однако и оно позволяет довольно полно

проверить большую систему. Для больших систем полное систематическое испытание может занимать много времени.

Лучше запускать испытание накануне вечером или, в особых случаях, на выходные. Даже если испытание занимает длительное время, оно должно быть выполнено, по крайней мере, один раз, чтобы найти ошибки.

Любые найденные ошибки будут показаны в окне Эксперта. Чтобы получить более детальные сведения относительно ошибки, щелкните кнопкой Детали (Details). На экране появится другое окно с дополнительными деталями относительно ошибки. Некоторые ошибки позволяют получить данные, которые могут использоваться, чтобы автоматически повторно перезапустить правила.

Как только система была проверена, генерируется файл отчета. Этот файл носит название <системное имя>.err. Его можно просмотреть и отпечатать. Это можно сделать в RuleBook, щелкнув кнопкой Отчет (Report).

Файл Отчет содержит информацию относительно найденных ошибок, качеств, переменных, альтернатив и использования правил. Эта статистика может быть исследована относительно контекста системы, чтобы определить, имеют ли выданные сообщения смысл.

2.3. Сообщения об ошибках

- а) Incomplete Branche - незавершенные ветви;
- б) Text Factors Not Used - текстовые факторы не использованы;
- в) Variables Not Used - переменные не использованы;
- г) Text Factors Not Always Derived - фактор текста не всегда определен;
- д) Variables Not Always Derived - переменные не всегда определены;
- е) Multiple Factors Use - многократное использование фактора;
- ж) Input Produces No Output - вход не производит никакого вывода;
- з) Variable Assigned a Value Out of Range - переменная вне диапазона;
- и) Too Many Values for Text Factor - слишком большое количество величин;
- к) Loop Error - ошибка замкнутого контура.

Рассмотрим сообщения об ошибках подробнее.

а) "незавершенные ветви"

Некоторые ветви в дереве неполны. Они имеют часть "ЕСЛИ", но нет части "ТО". Эти узлы (в дереве решений) будут заканчиваться красным узлом "добавить фактор". Они также могут быть найдены, если щелкнуть кнопкой на сообщении "Неполные узлы". Появится окно со списком неполных ветвей в каждом дереве.

Как исправить: Щелкните на опцию ДЕТАЛИ (или дважды щелкните на ветвь). Будет показано дерево с ошибочной ветвью. Ошибочная ветвь будет выделена черным. Выделение черным цветом может быть выключено после выбора опции "Очистить выделенные узлы" ("Clear Highlighted Nodes" в меню Zoom("Изменить масштаб изображения")). Добавьте условия "ТО" к ошибочным ветвям или (если они для получения данных) можно соответственно добавить узлы "Игнорируйте ветвь". Узел "игнорирование ветви" будет предотвращать будущие предупреждения относительно узла, но не будет генерировать правило. Добавление данных к узлу добавляет ветви (это случается, если добавлены дополнительные узлы "ЕСЛИ"). При редактировании список проблемных ветвей в списке

неполных ветвей станет неправильным, так как он будет⁴ применяться к первоначальному варианту дерева. Исправлять проблемные ветви лучше, начиная с узлов, находящихся в основании дерева. Если нумерация ветвей в списке неполных ветвей становится неправильной, запустите вновь Системный контроль (System Check) и остановите после быстрого контроля. Быстрый контроль произойдет моментально и исправит нумерацию даже после того, как было добавлено несколько ветвей;

б) не используются текстовые факторы

Некоторые из текстовых факторов (качества), определенные в системе, не используются ни в каком дереве. Это скорее предупреждение, чем истинная ошибка, однако если фактор участвует в процессе принятия решений, он должен быть куда-нибудь включен.

Как исправить: Щелкните на "Неиспользуемые текстовые факторы" (Text Factors Not Used) и щелкните на ДЕТАЛИ (Details) или дважды щелкните на "Неиспользуемые текстовые факторы" (Text Factors Not Used), и будет показан список неиспользуемых качеств. Так как проблема состоит в том, что эти качества не использовались, система не может обеспечить вас какими-либо дополнительными данными;

в) неиспользованные переменные

Некоторые из переменных, определенные в системе, не используются ни в каком дереве. Это скорее предупреждение, чем истинная ошибка, однако если переменная участвует в процессе принятия решений, она должна быть куда-нибудь включена.

Как исправить: Щелкните на "Неиспользуемые переменные" (Variables Not Used) и щелкните на ДЕТАЛИ (Details) или дважды щелкните на "Неиспользуемые переменные" (Variables Not Used), и список неиспользуемых переменных будет показан. Так как проблема состоит в том, что эти переменные не использовались, система не может обеспечить вас какими-либо дополнительными данными;

г) фактор текста не всегда определен

Некоторые из текстовых факторов иногда получаются в результате работы правил, а иногда запрашиваются у пользователя. Это скорее предупреждение, чем истинная ошибка, так как часть данных может быть получена только для некоторых комбинаций входов.

Как исправить: Щелкните на "Фактор текста не всегда определен" (Text Factors Not Always Derived) и щелкните на ДЕТАЛИ (или дважды щелкните на "Фактор текста не всегда установлен"), и вам будет представлен список качеств с этой ошибкой. После качества следует список входов, где качество запрашивается у конечного пользователя;

д) переменные не всегда определены

Некоторые из переменных иногда получаются в результате работы правил, а иногда запрашиваются у пользователя. Это скорее предупреждение, чем истинная ошибка, так как часть данных может быть получена только для некоторых комбинаций входов.

Как исправить: Щелкните на "Переменные не всегда определены" (Variables Not Always Derived) и щелкните на ДЕТАЛИ (или дважды щелкните на "Variables Not Always Derived"), и вы увидите список переменных с этой проблемой. После переменной следует список

входов, где переменная должна запрашиваться у конечного пользователя. (Случаи, где переменная могла бы быть получена, не внесены в список.);

е) многократное использование фактора

Сообщение говорит о том, что качество или переменная были использованы дважды в "ЕСЛИ"-части правила. Это ненормально, так как обычно устанавливается одно значение качества или переменной. Если качество или переменная требуются с различными величинами в одной и той же ветви, ветвь никогда не будет работать. Обычно это указывает на ошибку в логике дерева.

Как исправить: Отредактируйте дерево;

ж) из входа не следует никакого выхода

Некоторым комбинациям входов не соответствует ни одна из альтернатив. Конечный пользователь, отвечающий на вопросы таким образом, не получил никакой рекомендации.

Как исправить: Щелкните на "Input Produces No Output" и на ДЕТАЛИ (или дважды щелкните на "Input Produces No Output") и вы увидите список входов с этой проблемой;

з) переменная вне диапазона

Некоторые виды ввода устанавливают величину для переменной, которая является ниже нижнего предела или выше верхнего.

Как исправить: Для устранения ошибки получите список входов с этой проблемой и измените допустимый диапазон переменной;

и) слишком большое количество величин

Некоторые комбинации входов устанавливают большее чем максимальное число значений для качеств.

Как исправить: В диалоге для определения параметров качеств имеется опция для определения максимального числа значений, которые качество может иметь. Обычно оно установлено по умолчанию для всех величин, однако для отдельной величины его можно изменить. Для этого получите список качеств и поочередно исправьте их;

к) ошибка замкнутого контура

Ошибка замкнутого контура появляется, когда величина переменной или качества получены друг из друга таким путем, что цепь выводов идет по кругу. Например, величина [X] получена из величины [Y], но величина [Y] получена из [X]. Эта же ошибка может произойти с еще большим количеством пунктов в петле.

Как исправить: Последовательно щелкните на "Loop Error" и ДЕТАЛИ, и RuleBook найдет и автоматически разрушит петли. Однако обычно петли представляют собой ошибку в логике системы и должны быть исправлены вручную. Один раз сгенерировав правила, Вы можете проверить различные правила, чтобы найти проблему в логике замкнутого контура. Серьезные ошибки замкнутого контура могут создать проблемы, когда Вы пытаетесь их

исправить, так что всегда лучше сохранить систему правил перед попыткой исправить эти ошибки.

2.4. Степень уверенности

Разработанная система описывает процесс, приводящий к решению. В некоторых случаях каждая возможная комбинация данных позволяет сделать определенное заключение. Однако часто реальный мир не так взаимоувязан. Большинству экспертных систем приходится иметь дело с ситуациями, где набор предпосылок имеет определенное решение, но при этом предполагает наличие других решений.

Назначение коэффициентов уверенности различным решениям позволяет системе выбрать несколько возможных заключений и распределить их в порядке возрастания. Эти коэффициенты уверенности определяются путем сложения коэффициентов уверенности решений всех прав ил, входящих в цепочку рассуждений, приводящих к заключению. Выводится лишь определенное количество альтернатив с наивысшим коэффициентом уверенности.

RuleBook поддерживает три способа определения степени уверенности:

а) ДА/НЕТ.

Не используются сомнительные данные. Все альтернативы могут положительно отбираться или отклоняться. Величина выбора устанавливается в соответствии с первым правилом, которое назначает величина ДА(1) или НЕТ(0);

б) если 0, то точно НЕТ, если 10, то точно ДА. Величины от 1 до 9 усреднены, а величины 0 и 10 пороговые;

в) увеличение на 1/Уменьшение на 1.

При выборе способа определения степени уверенности для системы используют, по возможности, самый простой способ.

Используйте ДА/НЕТ способ, когда:

а) система не требует факторов уверенности;

б) альтернативы либо выбираются, либо отклоняются;

в) ранжирование отбираемых выборов не требуется.

Используйте 0-10 способ, когда:

а) по дереву нельзя сделать выбор с абсолютной уверенностью;

б) альтернативы будут выбираться в условиях независимости факторов;

в) некоторые правила должны допускать возможность назначать решению степень уверенности, равную 0 или 10;

г) большинство пользователей может легко определиться со степенью уверенности того или иного решения.

Используйте способ Увеличение на 1/Уменьшение на 1, когда:

а) система должна различать альтернативы, которые являются подобными;

б) предполагается, что в большинстве пунктов в решении очень много независимых факторов.

3. Описание работы с программным продуктом EXSYS RuleBook.

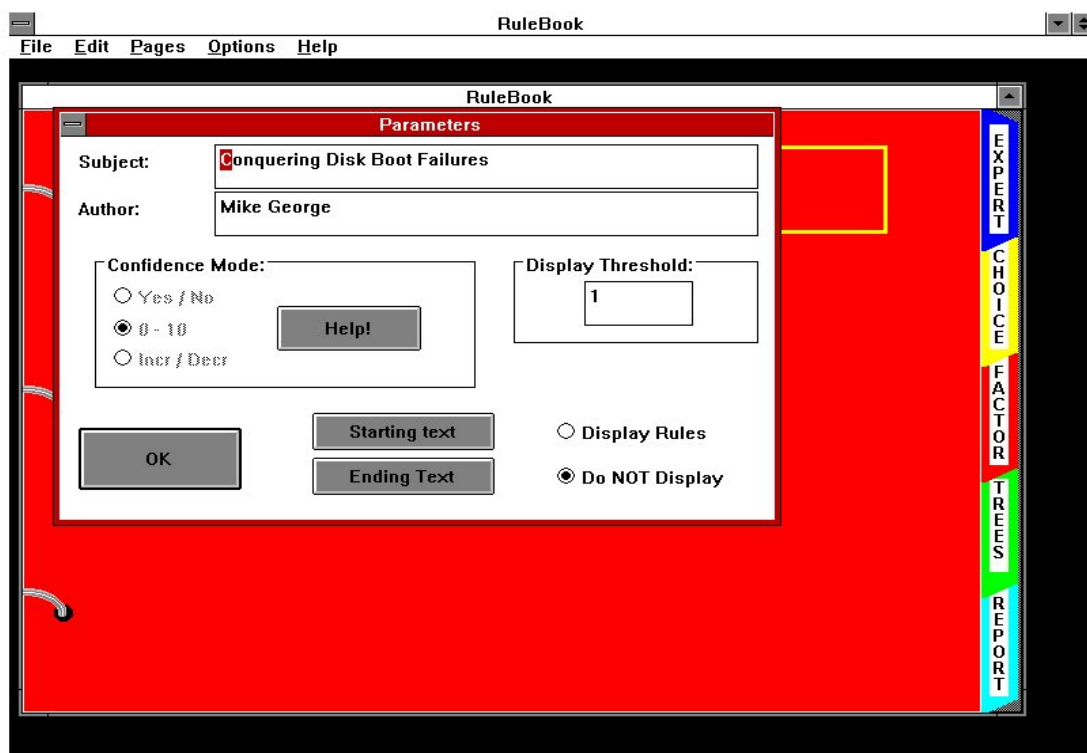
После запуска Rule Book на экране появится заставка с логотипом EXSYS Inc.

Доступными в этот момент являются лишь два пункта меню "File " и "Help" .

Пункт меню "File" дает возможность либо открыть уже существующий файл (Open),либо создать новый (New). Выбрав подпункт "New",пользователь получает возможность ввести имя файла, в котором будет содержаться создаваемая система (рисунок1).



После ввода имени файла на экране появится окно(рисунок2), при помощи которого разработчик системы создает своего рода титульный лист, называя объект (Subject), над которым работает и себя (Author).



В этом окне пользователь должен также задать способ определения степени уверенности

"Yes/No"-если система должна дать односложный ответ на определенный вопрос, например, ответом на вопрос "Пойдет ли дождь?",будет либо "Да",либо "Нет ".

"0-10"-система должна выбрать одно или несколько решений из набора возможных для каждого из которых можно определить степень уверенности в его точности.

"Incr/Decr"-система должна дать положительное либо отрицательное заключение. При движении системы к заключению она добавляет, либо вычитает единицу от базового значения. В результате если базовое значение отрицательное, то и заключение отрицательное, а если положительное ,то и заключение положительное.

Используя кнопки "Starting text" "Ending text ",разработчик может задать текст, который система будет выдавать на экран в начале и в конце своей аботы.

Введя всю вышеперечисленную информацию разработчик получает возможность непосредственно приступить к созданию базы знаний.

Первым этапом создания БЗ является описание всех возможных заключений которые мог бы сделать эксперт, решая поставленную задачу. Сразу же после заполнения титульного листа и нажатия клавиши "OK" на экране появится окно для введения этих заключений(рисунок3).

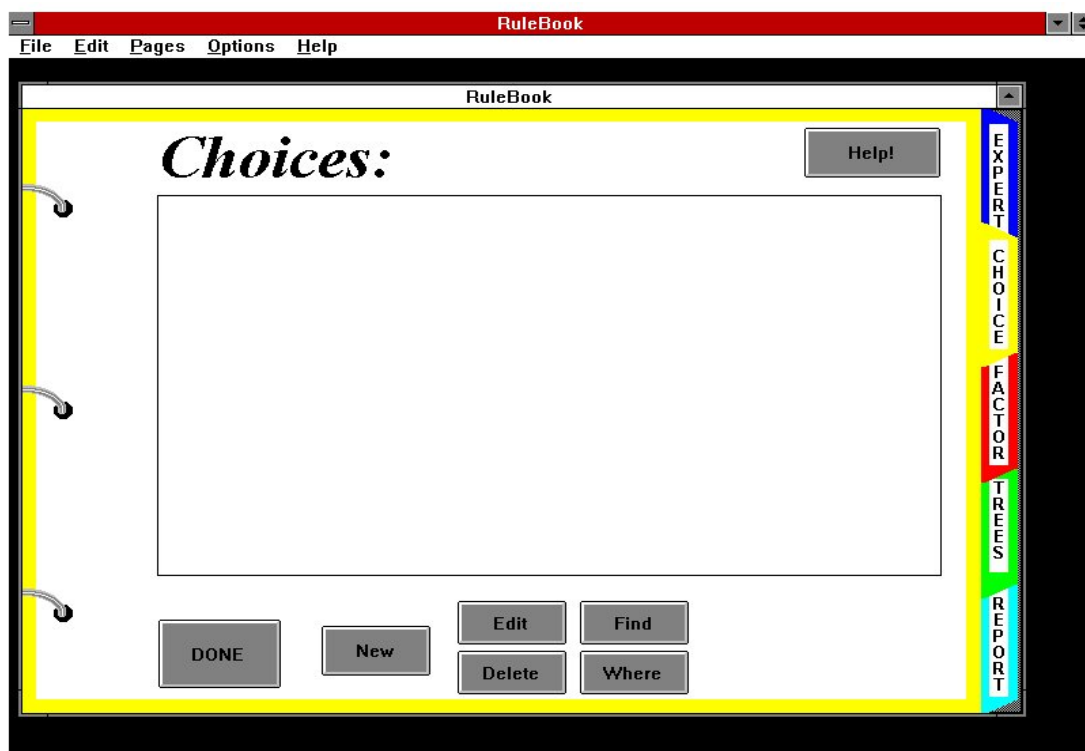


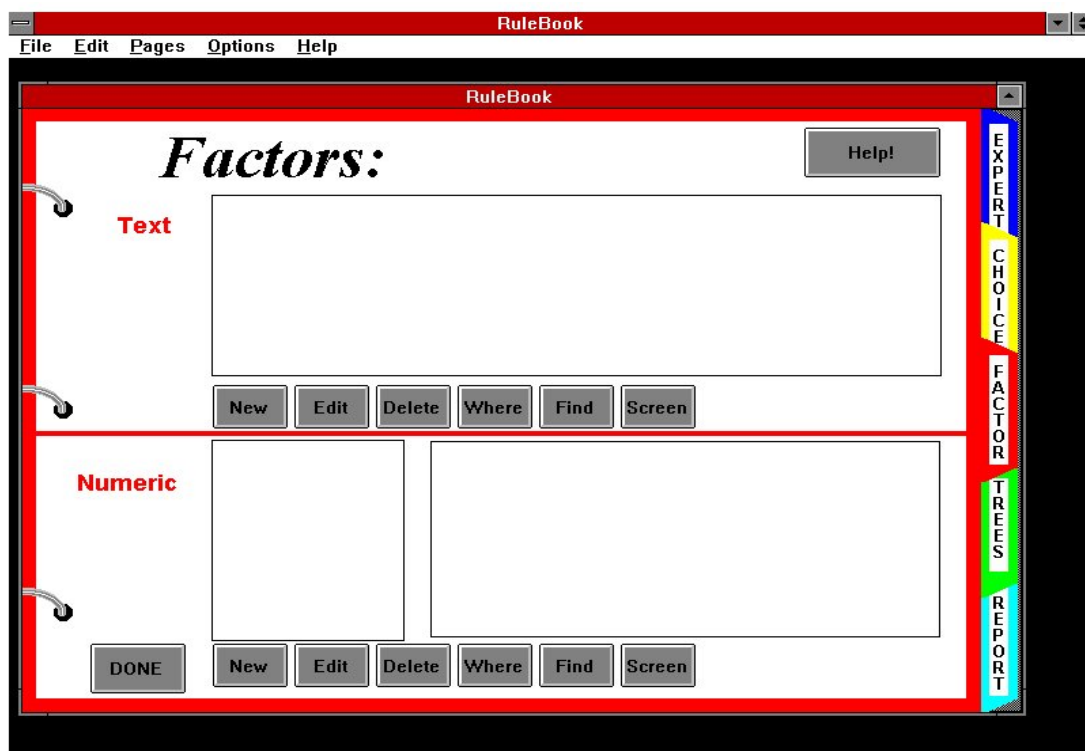
Таблица 1 Функции кнопок в окне "CHOICES"

Кнопка	Выполняемое действие
NEW	Создать новое заключение
EDIT	Отредактировать текущее заключение
DELETE	Удалить текущее заключение
FIND	Найти заключение
WHERE	Где используется текущее заключение
DONE	Закончить текущее заключение

Используя перечисленные в таблице 1 кнопки, разработчик создает список заключений, а затем нажав кнопку "DONE", переходит к следующему этапу разработки БЗ-вводу факторов . Фактор- это критерий оценки исследуемого предмета ,например ,если в качестве исследуемого предмета используется автомобиль(т. е. при помощи данной ЭС пользователь будет выбирать наиболее подходящий для него автомобиль),то список факторов будет содержать цвет , форму,мощность двигателя , экономичность и т. д .

Ввод списка факторов осуществляется из окна "FACTORS"(рисунок4) при помощи кнопок, описанных в таблице 2 :

Кнопка	Выполняемое действие
NEW	Создать новый фактор
EDIT	Отредактировать текущий фактор
DELETE	Удалить текущий фактор
WHERE	Где используется текущий фактор
FIND	Найти фактор
SCREEN	Создать экран для ввода данных



Окно "FACTORS" разделено на две части:

TEXT-для введения текстовых факторов;

NUMERIC-для введения числовых факторов.

После нажатия кнопки "NEW" на экране появится окно, изображенное на рисунке5 . Поле "Qualifier" предназначено для введения наименований факторов (например , "цвет автомобиля "), поле "Value" -для введения их характеристик (" белый ", " красный" и т. д.).

Add / Edit Qualifier

Qualifier:1 NEW QUALIFIER Name: **Help!**

Qualifier:

Value:

Edit **Replace** **Undo Edit** **Add** **Delete**

OK ☐ **Display at end** **Custom Screen**

Cancel ☐ **Limit input values to:** **Custom Help**

☐ **Default value:**

Для введения числовых факторов в RuleBook используется окно изображенное на рисунке 6.

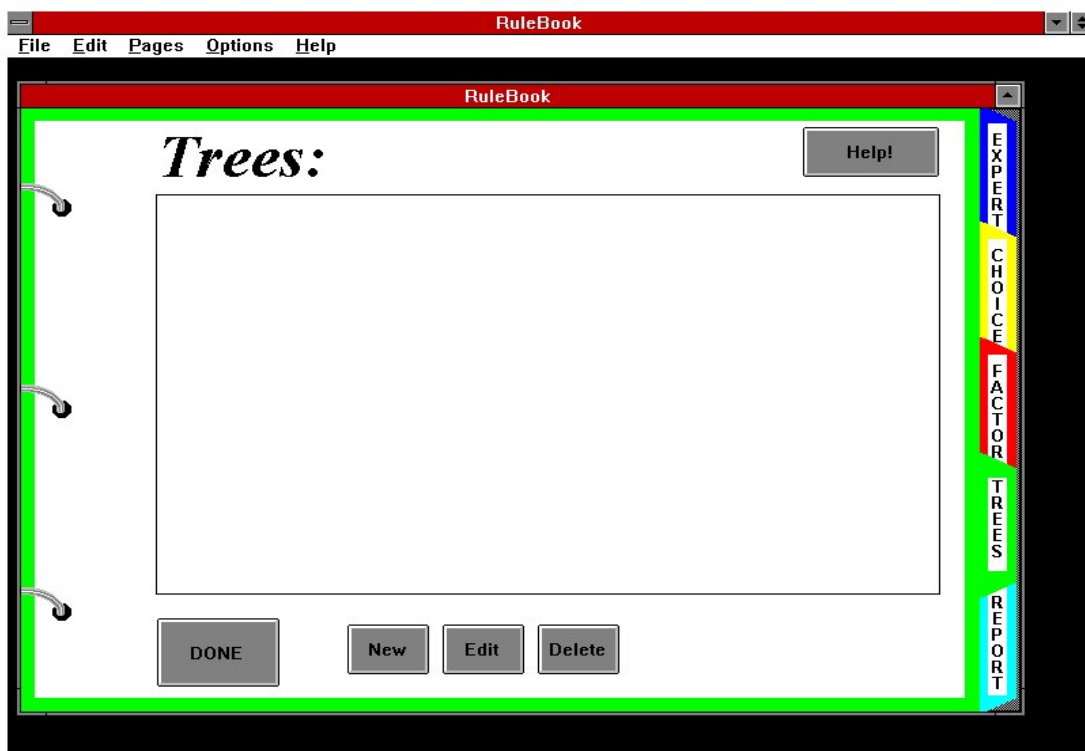
В поле "Prompt" разработчик задает название фактора (например, "мощность двигателя"), в поле "Lower Limit" и "Upper Limit"-соответственно нижний и верхний пределы фактора, в поле "Ranges"- допустимый диапазон ,а в поле "Range Break Pts."-точки деления допустимого диапазона (например ,допустимый диапазон мощностей двигателя можно разделить на три части: маломощные, среднемощные и высокомоощные)

Следующим этапом создания БЗ является построение дерева правил на основе введенных ранее факторов их значений и заключений. Используя построенное разработчиком дерево, RuleBook автоматически генерирует все возможные правила.

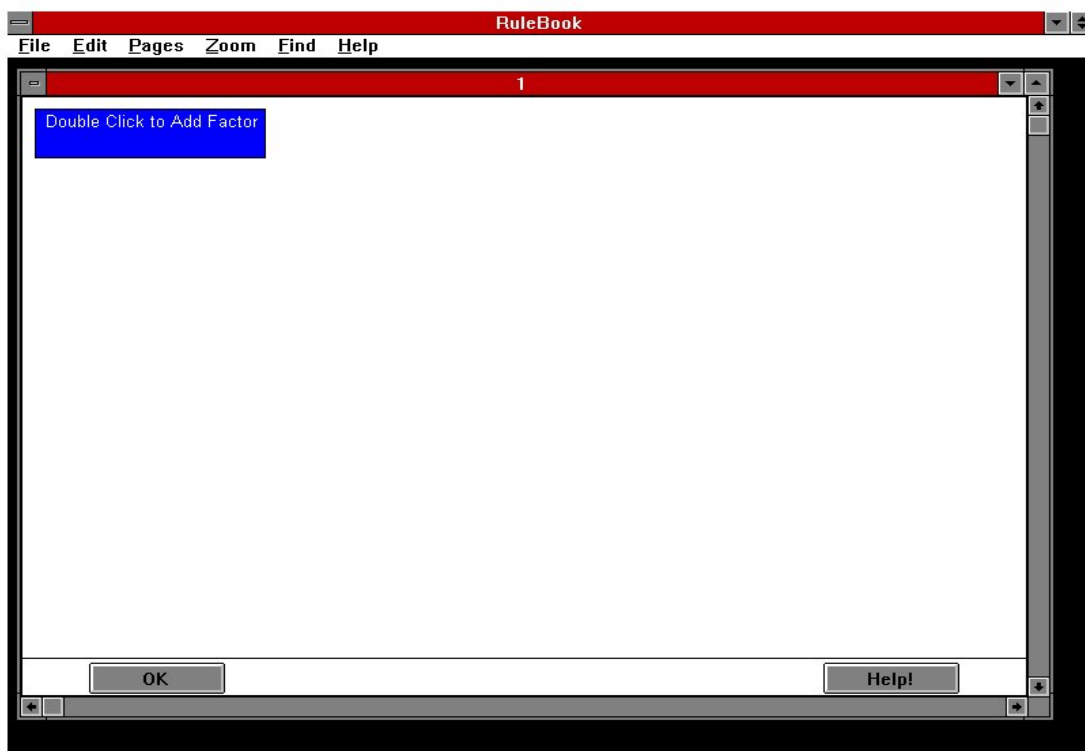
Построение дерева правил начинается из окна "TREES " (рисунок7) при помощи кнопок , описанных в таблице 3.

Таблица 3. Функции кнопок в окне "TREES"

Кнопка	Выполняемое действие
NEW	Создание нового дерева
EDIT	Редактирование текущего дерева
DELETE	Удаление текущего дерева



Нажав кнопку "NEW" и введя название нового дерева, разработчик получает доступ к окну, изображенному на рисунке 8.



Создание узлов дерева осуществляется из окна "Add Factor" (рисунок9), вызвать которое можно двойным щелчком мыши на узле, обозначенном красным цветом.

Add Factor

Add IF Factor:

Text Numeric

Expression

Add THEN:

Choice

Text Numeric

Message Comment

Ignore Branch

Move to Upper Left

Paste

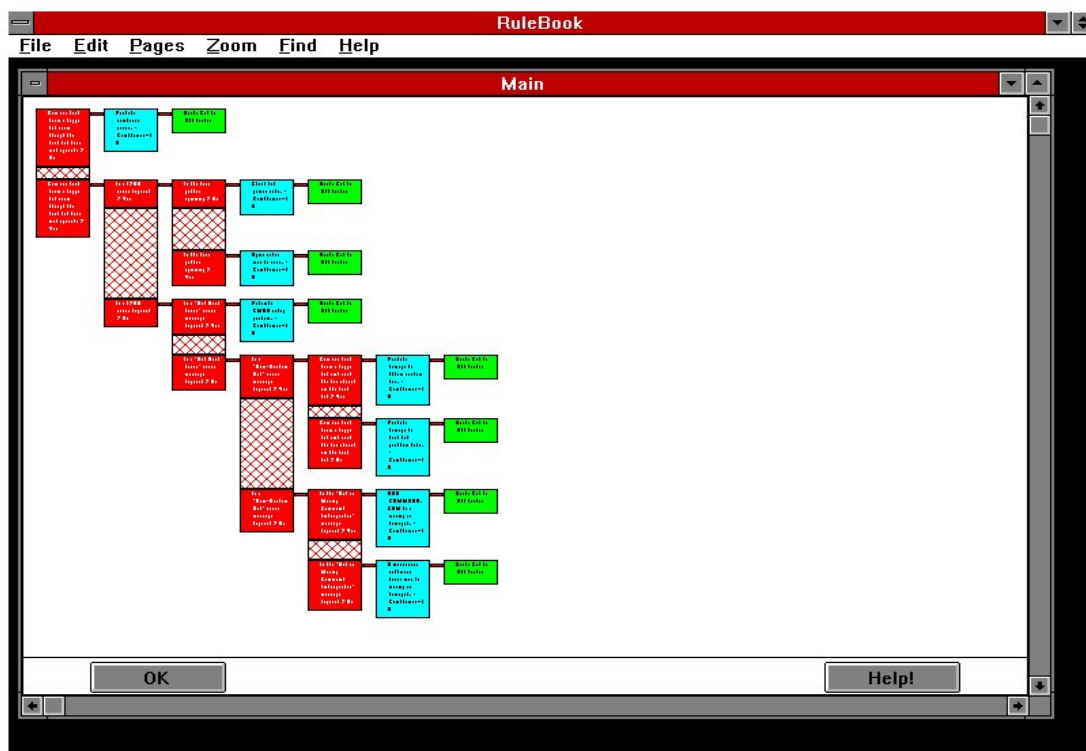
Cancel Help!

В данном окне имеются две основные области:

Add IF Factor Содержит кнопки, позволяющие добавлять в дерево узлы ветвления;

Add THEN Содержит кнопки, позволяющие добавлять в дерево узлы заключений.

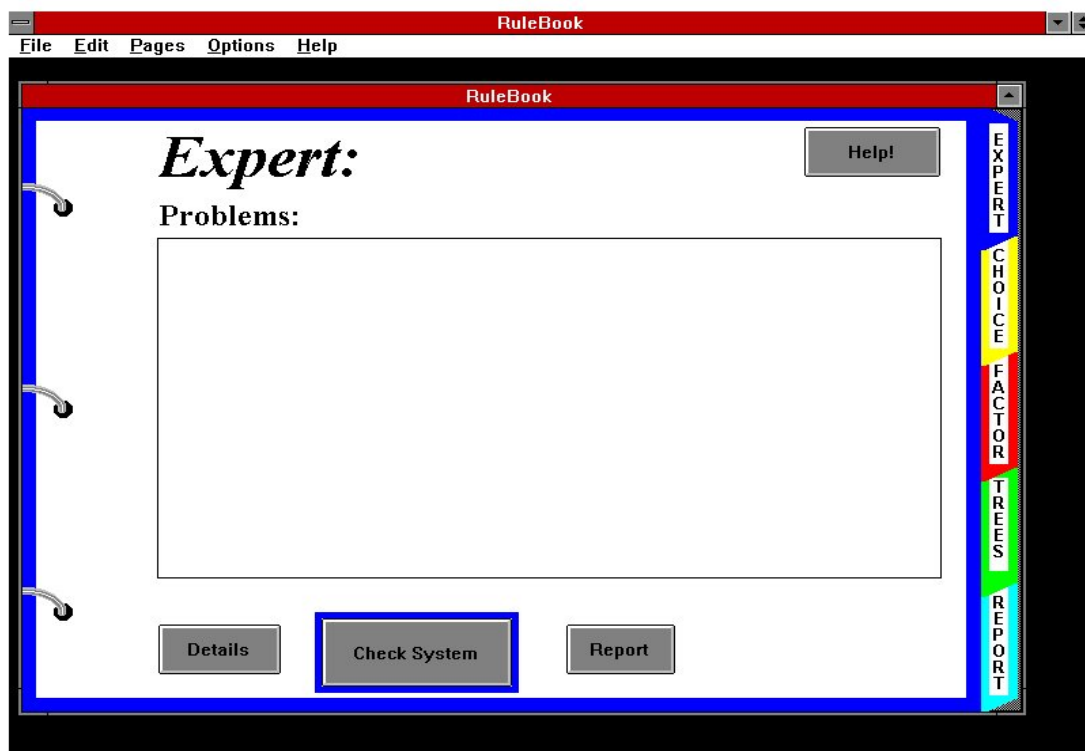
Дерево решений строится соответственно логике принятия решения применяемой экспертом на практике. Например, дерево решений, изображенное на рисунке 10, предназначенное для выдачи заключения о наиболее вероятной поломке компьютера.



Построив таким образом дерево решений, разработчик получает БЗ, проверить которую можно при помощи "Эксперта"(EXPERT) (рисунок11).

Таблица 4. Функции кнопок в окне"EXPERT"

Кнопка	Выполняемое действие
Check system	Запуск проверки БЗ на наличие ошибок
Details	Пояснение выводимых сообщений
Report	Создание отчета о результатах проверки



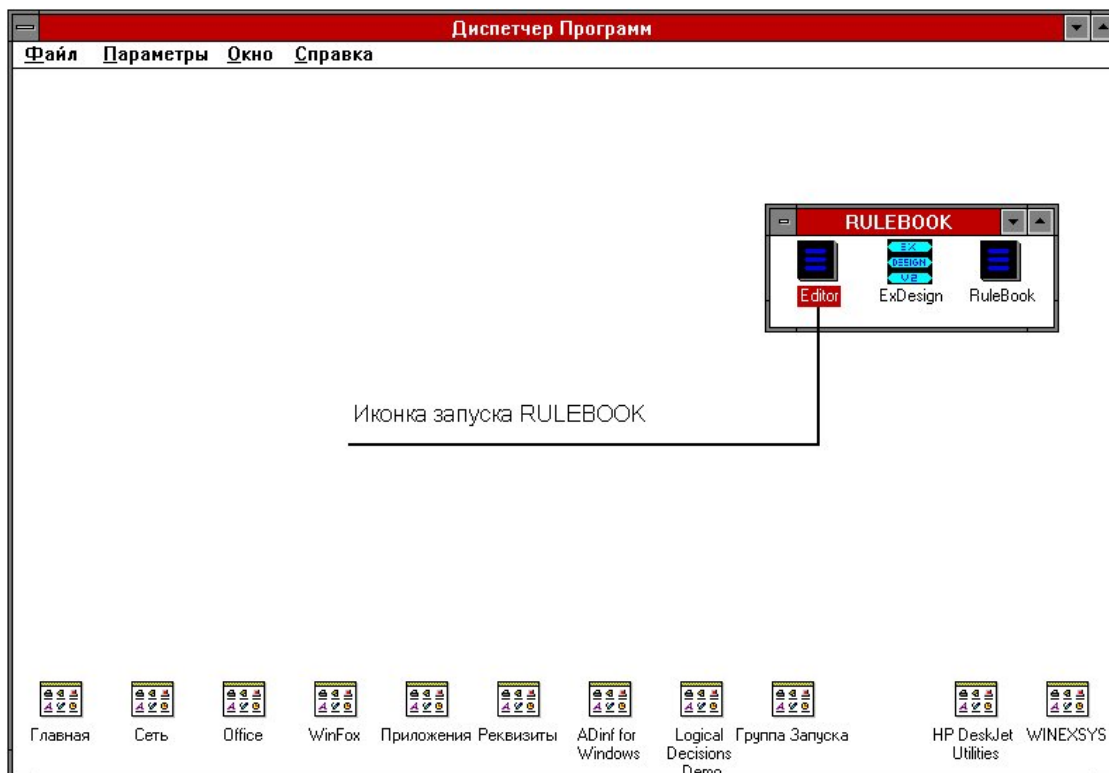
4. Порядок выполнения работы.

В качестве исходной информации к заданию выступает конкретно определенная преподавателем предметная область, для которой создается ЭС.

4.1. Создание ЭС.

4.1.1. Изучить методику построения ЭС при помощи приведенную в п.2;

4.1.2. Загрузить систему дважды щелкнув по иконке (рисунок12);



4.1.3. Провести анализ данных полученных в п 4

- определить цели
- определить критерии оценки (составить список)
- составить список значений, которые может принимать каждый из определенных выше критериев
- выбрать способ определения степени уверенности

4.1.4. Построить экспертную систему на основе проведенного выше анализа, работа которой отражала бы процесс принятия решения экспертом.

Примечание: Если возникают вопросы при работе с программными продуктами воспользуйтесь контекстной подсказкой, нажав клавишу F1. Также в качестве эксперта может выступать преподаватель.

5. Контрольные вопросы

1. Укажите цель построения экспертных систем
2. Перечислите этапы построения экспертных систем
3. Перечислите возможные способы определения степени уверенности используемые в Rule Book.
4. Что такое база знаний?
5. Каким образом дерево решений связано с базой знаний?
6. Кто такой эксперт?